universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Introdução a Programação para Games

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Teste de Mesa

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

Teste de Mesa

- **Definição**
  - Consiste em simular a execução de um algoritmo sem utilização do computador
  - Técnica para verificar se um determinado algoritmo leva ao resultado esperado

Teste de Mesa

- **Processo**
  - **Identifique as variáveis presentes no algoritmo**
  - **Crie uma tabela:**
    - **Cada coluna representa uma variável**
    - **Cada linha representa uma mudança de valor em uma ou mais variáveis**
    - **Opcionalmente, é possível marcar a linha do código que foi processada para facilitar a interpretação**

Teste de Mesa

- **Exemplo**

```
int valor = 5;
int resultado = 1;

for (int n = 1; n <= valor; n++) {
    resultado = resultado * n;
}
println(resultado);
```

LINHA	valor	resultado	n
1	5	-	-
2	5	1	-
3	5	1	1
4	5	1	1
3	5	1	2

Teste de Mesa

- **Exercício**
  - Continue a realizar o teste de mesa do exercício anterior
  - Qual valor será impresso no console?
  - O que o algoritmo calcula?

Teste de Mesa

- **Exercício**
  - **Realize o teste de mesa do seguinte algoritmo**

```
int valor = 1234;
int resultado = 0;

while (valor >= 1) {
  resultado += valor % 10;
  valor = valor / 10;
}
println(resultado);
```

universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Funções

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

Funções

- **Definição**
  - Recurso da programação que permite isolar partes do código e facilita a repetição
  - Algumas funções são predefinidas pela linguagem, exemplos: print(), random()

Funções

- **Definição**
  - A sintaxe completa para criação de uma função segue o padrão:
    - tipo nome(parâmetros) { comandos }
  - O tipo determina o tipo de dado do retorno da função, caso não tenha retorno o tipo será void
  - O nome representa o nome da função que é um identificador

Funções

- **Definição**
  - Os **parâmetros** são variáveis que precisam ter seus valores estabelecidos para a execução da função, pode não ter nenhum
  - Os **comandos** representam o trecho de código a ser executado quando a função for chamada

Funções

- **Definição**
  - **ATENÇÃO, variáveis criadas em uma função só existem dentro desta função**
  - **Para uma variável existir em mais de uma função é necessário que ela seja uma variável GLOBAL, ou seja, que foi declarada fora das funções**

Funções

- **Utilização**
  - **Para criar uma função o primeiro passo é definir a assinatura da função, na qual inclui seu nome**
  - **Exemplo:**

```
void funcao()
{
    //comandos
}
```

Funções

- **Utilização**
  - O passo seguinte é definir os comandos que serão executados quando a função é chamada, definindo o seu **corpo**
  - Exemplo:

```
void funcao()
{
    println("Executou");
}
```

Funções

- **Utilização**
  - **Por fim, para executar a função basta usar seu nome seguido de parênteses, o que denominamos de chamar a função**
  - **Exemplo:**

```
funcao();

void funcao()
{
    println("Executou");
}
```

universidade anhembi morumbi
Mundialmente criativa e inovadora
Laureate International Universities®

Processing

Prof. Alexandre Nascimento Tomoyose

2º semestre 2016

Processing

- **Funções no Processing**
  - **Existem duas formas de utilizar o processing com ou sem funções**
  - **Se for utilizar com funções é obrigatório o uso de duas funções especiais: setup() e draw()**

Processing

- **Funções no Processing**
  - **A função setup é executada apenas uma vez no início**
  - **A função draw é executada continuamente, uma vez a cada frame**

Processing

- **Exercício**
  - Crie um programa que desenha um retângulo por frame com posição, tamanho e cor aleatória
  - Transforme o código em uma função e chame a função a partir do draw

Processing

- **Exercício**
  - **Faça 3 funções diferentes, uma que cria retângulo, uma que cria elipse e outra que cria triângulo, todos com posição, tamanho e cor aleatórios**
  - **Altere o draw para fazer um sorteio todo frame que resulte em 1/3 de chance de criar um retângulo e 1/3 de criar uma elipse e 1/3 de criar um triângulo**